In the Claims:

Please amend claims 1, 5 and 16. The claims are as follows:

1. (Currently Amended) An instruction buffer comprising:

a physical memory array partitioned into multiple physical and identical memory sub-arrays arranged in sequential order from a first memory sub-array to a last memory sub-array, each memory sub-array having multiple physical instruction entry positions and adapted to store a different instruction of a set of concurrent instructions in a single instruction entry position of any one of said memory sub-arrays, said set of concurrent instructions arranged in sequential order from a first instruction to a last instruction.

2. (Original) The instruction buffer of claim 1, wherein:

each instruction of said set of concurrent instructions is stored in a different memory sub-array and the set of concurrent instructions may wrap from said last memory sub-array to said first memory sub-array.

3. (Original) The instruction buffer of claim 2, wherein when said set of concurrent instruction wraps around, each instruction that is wrapped is written to an instruction entry position in a corresponding memory sub-array that is one instruction entry position higher in a memory sub-array corresponding to each instruction that is not wrapped except an instruction wrapped from the last instruction entry position of the last memory sub-array wraps to the first instruction entry position of the first memory sub-array.

4. (Original) The instruction buffer of claim 1, wherein each memory sub-array comprises single write port and single read port memory cells.

5. (Currently Amended) The instruction buffer of claim 1, wherein a number of write ports is less than a number of instructions to be written into said memory array and a number of read ports is less than a number of instructions to be read out of said memory array.

6. (Original) The instruction buffer of claim 5, wherein:

each instruction entry position is defined by a physical instruction entry position and a corresponding logical instruction entry position, each logical instruction entry position in a particular memory sub-array is a fixed number higher than an immediately previous logical instruction entry position in said particular memory sub-array;

said physical instruction entry positions in each memory sub-array are one logical instruction entry position higher than corresponding physical entry positions of a immediately previous memory sub-array, the first physical instruction entry position and first logical instruction entry position of a first memory sub-array being the same; and

each said instruction of said set of concurrent instructions is stored in consecutive logical instruction entry positions of said memory array.

7. (Original) The instruction buffer of claim 6, further including:

a rotator multiplexer adapted to receive said set of concurrent instructions and direct each instruction of said set of concurrent instructions to an entry position in different consecutive memory sub-arrays;

an output multiplexer adapted to order a sequence of instructions read out of said memory array to match the order of instructions in said set of concurrent instructions;

a write address decoder adapted to determine a write address of a wordline of said memory array to which said first instruction of said set of instructions will be written; and

a read address decoder adapted to determine a read address of a wordline of said memory array from which a first instruction of a group of instructions will be read from.

8. (Original) The instruction buffer of claim 7, wherein:

a maximum number of instructions in said set of concurrent instructions is equal to N instructions;

said rotator multiplexer comprises N, N:1 multiplexers, each multiplexer adapted to select one of said N instructions and couple the selected instruction to one of said memory sub-arrays.

9. (Original) The instruction buffer of claim 7, wherein:

a maximum number of instructions in said set of concurrent instructions is equal to N instructions;

said rotator multiplexer comprises:

N, (N/2):1 first multiplexers arranged in pairs, each first multiplexer adapted to select one instruction from a different sequence of N/2 different sequences of instructions of said set of concurrent instructions;

N, (2:1) second multiplexers arranged in pairs, each first multiplexer adapted to select one instruction from a corresponding pair of said first (N/2):1 multiplexers and couple the selected instructions to the inputs of different memory sub-arrays.

10. (Original) The instruction buffer of claim 9, wherein:

a number of said physical entry positions in each said memory sub-array is equal to Q;

N is equal to 8 and Q is equal to 8;

the first instruction in said set of concurrent instructions has a physical instruction entry position defined by a 6-bit address, a first set of 3-bits of said 6-bit address defining a physical entry position in each memory sub-array and a second set of 3-bits of said 6-bit address defining a position of said sub-array in said memory array; and

said write address decoder comprises:

a write wordline decoder adapted to generate an 8-bit address of said first set of 3-bits and a 1-bit to the left shifted 8-bit address of said 8-bit address;

seven 2:1 address multiplexers, an output of each address multiplexer coupled to a corresponding wordline select of seven sequential memory sub-arrays beginning with said first memory sub-array, a first input of each address multiplexer coupled to said 8-bit address, a second input of each address multiplexer coupled to said shifted 8-bit address and the select input of each address multiplexer coupled to a logic circuit, said logic circuit coupled to said second set of 3-bits; and

said 8-bit address coupled to a wordline select of said last memory sub-array.

11. (Original) The instruction buffer of claim 9, wherein:

a number of said physical entry positions in each said memory sub-array is equal to Q;

a maximum number of instructions in said set of concurrent instructions to be read out of said memory array concurrently is equal to M;

N is equal to 8 and Q is equal to 8;

the first instruction in said set of concurrent instructions has a physical instruction entry position defined by a 6-bit address, a first set of 3-bits of said 6-bit address defining a physical entry position in each memory sub-array and a second set of 3-bits of said 6-bit address defining a position of said sub-array in said memory array; and

said read address decoder comprises:

a read wordline decoder adapted to generate an 8-bit address of said first set of 3-bits and a 1-bit to the left shifted 8-bit address of said 8-bit address;

(M-1) 2:1 address multiplexers, an output of each address multiplexer coupled to a corresponding wordline select of (M-1) consecutive memory sub-arrays starting with said first memory sub-array, a first input of each address multiplexer coupled to said 8-bit address, a second input of each address multiplexer coupled to said shifted 8-bit address and the select input of each address multiplexer coupled to one bit of said second set of 3-bits; and

said 8-bit address coupled to a wordline select of the each memory sub-arrays not coupled to an output of one of said (M-1) 2:1 multiplexers.

12. (Original) The instruction buffer of claim 11, wherein M is equal to 5.

13. (Original) The instruction buffer of claim 7, wherein:

a maximum number of instructions to be read out of said memory array concurrently is M instructions; and

said output multiplexer comprises M, N:1 multiplexers.

14. (Original) The instruction buffer of claim 7, wherein:

a maximum number of instructions in said set of concurrent instructions is equal to N instructions;

a number of said memory sub-arrays in said memory array is N;

a number of said physical entry positions in each said memory sub-array is equal to Q; and

Q may or may not be equal to N.

15. (Original) The instruction buffer of claim 1, wherein each memory sub-array is selected from the group consisting of static random access memory arrays, dynamic random access memory arrays, latch arrays, or a register array.

16. (Currently Amended) A method of buffering instructions for a processor, comprising:

providing a physical memory array partitioned into multiple physical and identical memory sub-arrays arranged in sequential order from a first memory sub-array to a last memory sub-array, each memory sub-array having multiple physical instruction entry positions and adapted to store a different instruction of a set of concurrent instructions in a single instruction entry position of any one of said memory sub-arrays, said set of concurrent instructions arranged in sequential order from a first instruction to a last instruction; and

writing and reading instructions of said set of concurrent instructions to and from said memory array.

17. (Original) The method of claim 16, wherein:

each instruction of said set of concurrent instructions is stored in a different memory sub-array and the set of concurrent instructions may wrap from said last memory sub-array to said first memory sub-array.

18. (Original) The method of claim 17, wherein when said set of concurrent instruction wraps around, each instruction that is wrapped is written to an instruction entry position in a corresponding memory sub-array that is one instruction entry position higher in a memory sub-array corresponding to each instruction that is not wrapped except an instruction wrapped from the last instruction entry position of the last memory sub-array wraps to the first instruction entry position of the first memory sub-array.

19. (Original) The method of claim 16, wherein each memory sub-array comprises single write port and single read port memory cells.

20. (Original) The method of claim 16, wherein a number of write ports is less than a number of instructions to be written into said memory array and a number of read ports is less than a number of instructions to be read out of said memory array.

21. (Original) The method of claim 20, wherein:

each instruction entry position is defined by a physical instruction entry position and a corresponding logical instruction entry position, each logical instruction entry position in a particular memory sub-array is a fixed number higher than an immediately previous logical instruction entry position in said particular memory sub-array;

said physical instruction entry positions in each memory sub-array are one logical instruction entry position higher than corresponding physical entry positions of a immediately previous memory sub-array, the first physical instruction entry position and first logical instruction entry position of a first memory sub-array being the same; and

each said instruction of said set of concurrent instructions is stored in consecutive logical instruction entry positions of said memory array.

22. (Original) The method of claim 21, further including:

providing a rotator multiplexer adapted to receive said set of concurrent instructions and direct each instruction of said set of concurrent instructions to an entry position in different consecutive memory sub-arrays;

providing an output multiplexer adapted to order a sequence of instructions read out of said memory array to match the order of instructions in said set of concurrent instructions;

providing a write address decoder adapted to determine a write address of a wordline of said memory array to which said first instruction of said set of instructions will be written; and

providing a read address decoder adapted to determine a read address of a wordline of said memory array from which a first instruction of a group of instructions will be read from..

23. (Original) The method of claim 22, wherein:

a maximum number of instructions in said set of concurrent instructions is equal to N instructions;

said rotator multiplexer comprises N, N:1 multiplexers, each multiplexer adapted to select one of said N instructions and couple the selected instruction to one of said memory sub-arrays.

24. (Original) The method of claim 22, wherein:

a maximum number of instructions in said set of concurrent instructions is equal to N instructions;

said rotator multiplexer comprises:

N, (N/2):1 first multiplexers arranged in pairs, each first multiplexer adapted to select one instruction from a different sequence of N/2 different sequences of instructions of said set of concurrent instructions;

N, (2:1) second multiplexers arranged in pairs, each first multiplexer adapted to select one instruction from a corresponding pair of said first (N/2):1 multiplexers and couple the selected instructions to the inputs of different memory sub-arrays.

25. (Original) The method of claim 24, wherein:

a number of said physical entry positions in each said memory sub-array is equal to Q;

N is equal to 8 and Q is equal to 8;

the first instruction in said set of concurrent instructions has a physical instruction entry position defined by a 6-bit address, a first set of 3-bits of said 6-bit address defining a physical

entry position in each memory sub-array and a second set of 3-bits of said 6-bit address defining a position of said sub-array in said memory array; and

said write address decoder comprises:

a write wordline decoder adapted to generate an 8-bit address of said first set of 3-bits and a 1-bit to the left shifted 8-bit address of said 8-bit address;

seven 2:1 address multiplexers, an output of each address multiplexer coupled to a corresponding wordline select of seven sequential memory sub-arrays beginning with said first memory sub-array, a first input of each address multiplexer coupled to said 8-bit address, a second input of each address multiplexer coupled to said shifted 8-bit address and the select input of each address multiplexer coupled to a logic circuit, said logic circuit coupled to said second set of 3-bits; and

said 8-bit address coupled to a wordline select of said last memory sub-array.

26. (Original) The method of claim 24, wherein:

a number of said physical entry positions in each said memory sub-array is equal to Q;

a maximum number of instructions in said set of concurrent instructions to be read out of said memory array concurrently is equal to M;

N is equal to 8 and Q is equal to 8;

the first instruction in said set of concurrent instructions has a physical instruction entry position defined by a 6-bit address, a first set of 3-bits of said 6-bit address defining a physical entry position in each memory sub-array and a second set of 3-bits of said 6-bit address defining a position of said sub-array in said memory array; and

said read address decoder comprises:

a read wordline decoder adapted to generate an 8-bit address of said first set of 3-bits and a 1-bit to the left shifted 8-bit address of said 8-bit address;

(M-1) 2:1 address multiplexers, an output of each address multiplexer coupled to a corresponding wordline select of (M-1) consecutive memory sub-arrays starting with said first memory sub-array, a first input of each address multiplexer coupled to said 8-bit address, a second input of each address multiplexer coupled to said shifted 8-bit address and the select input of each address multiplexer coupled to one bit of said second set of 3-bits; and

said 8-bit address coupled to a wordline select of the each memory sub-arrays not coupled to an output of one of said (M-1) 2:1 multiplexers.

27. (Original) The method of claim 26, wherein M is equal to 5.

28. (Original) The method of claim 22, wherein:

a maximum number of instructions to be read out of said memory array concurrently is M instructions; and

said output multiplexer comprises M, N:1 multiplexers

29. (Original) The method of claim 22, wherein:

a maximum number of instructions in said set of concurrent instructions is equal to N instructions;

a number of said memory sub-arrays in said memory array is N;

a number of said physical entry positions in each said memory sub-array is equal to Q; and

Q may or may not be equal to N.

30. (Original) The method of claim 16, wherein each memory sub-array is selected from the group consisting of static random access memory arrays, dynamic random access memory arrays, latch arrays, or a register array.